REMARKS

Claims 1-4 were pending in the present application and stand rejected. Of these, claim 1 is independent. The applicant has amended claims 1, 3, and 4. The applicant has added new claims 5- 50, of which claims 5 and 28 are independent claims. The applicant thanks the Examiner for the interview conducted on April 9, 2003 and submits with this response an interview summary.

The Examiner has objected to the oath/declaration because it does not identify the city and state of residence of the first listed inventor. The applicant submits with this response an application data sheet. The examiner has objected to the abstract because the applicant has included on the same page a document number the applicant uses for administrative purposes. The applicant has removed the document number. The Examiner has objected to claim 4 as being informal. The applicant has amended claim 4 to correct the informality.

Claim 1 stands rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 5,999,173 to Ubillos ("Ubillos"). The respectfully traverses the rejection and has amended the claim for clarity. Claim 1 now recites: "A computer program product, tangibly stored on a computer-readable medium, for calculating the validity of a cached frame of a movie in a movie compositing system, comprising instructions operable to cause a programmable processor to:

maintain a global editing timestamp that is updated with each editing operation performed by the system, the global editing timestamp representing an edit sequence position in a sequence of edits made to the movie;

establish an interval list for each node in a compositing tree defining a movie, each node having a timeline that maps to a master timeline of the movie, an interval list for a node including, for each interval in the node's timeline, a single global editing timestamp;

update the interval list for a node when the node is edited; and

use the interval list for a first node to evaluate the validity of a cached frame for a particular interval of the master timeline, the cached frame having been produced by compositing the first node in the compositing tree, the evaluation being performed by (a) comparing (i) a global editing timestamp associated with the cached frame with (ii) the global editing timestamps

of intervals in the interval list that map to at least a portion of the interval of the master timeline, and (b) treating the cached frame as invalid if any of the intervals' global editing timestamps is later than the global editing timestamp associated with the cached frame."

Claim 1 includes elements not disclosed by Ubillos. For example, claim 1 recites "maintain a global editing timestamp that is updated with each editing operation performed by the system, the global editing timestamp representing an edit sequence position in a sequence of edits made to the movie." The applicant respectfully submits that the claimed editing timestamp is not disclosed or suggested by Ubillos. The Examiner contends that Ubillos discloses timelines and that a timeline of a movie discloses or suggests the claimed editing timestamp. The applicant agrees that Ubillos discloses timelines but must disagree that timelines disclose or suggest the claimed editing timestamp. The applicant has defined, in the specification and now in the amended claim 1, the term "editing timestamp" to represent an edit sequence position of an edit in a sequence of edits. An edit having an edit sequence position of four, for example, was the fourth edit made. A timeline of a movie is different from an edit sequence position of an edit that was made to the movie. A timeline of a movie, also referred to as a master timeline or the run time, does not indicate the edit sequence positions of edits. The time values included in a timeline indicate only when particular frames are displayed. Thus, a timeline does not disclose or suggest the claimed edit sequence position and Ubillos does not disclose or suggest the above cited element of claim 1. For at least this reason, the applicant respectfully submits that claim 1 and claims 2-4, which depend from claim 1, are in condition for allowance.

The applicant has added claims 5-50, of which claims 5 and 28 are independent. Claims 5 and 28 include elements that are analogous to the above cited element of claim 1 and the foregoing arguments apply with equal force to claims 5 and 28. The applicant respectfully submits that claims 5, 28, and their respective dependent claims are in condition for allowance.

In the claims:

Please amend the claims as follows:

---

1. (Currently amended) A computer program product, tangibly stored on a computer-readable medium, for calculating the validity of a cached frame of a movie in a movie compositing system, comprising instructions operable to cause a programmable processor to:

A1

maintain a global editing timestamp that is updated with each editing operation performed by the system, the global editing timestamp representing an edit sequence position in a sequence of edits made to the movie;

establish an interval list for each node in a compositing tree defining a movie, each node having a ~~range of possible times~~timeline that maps to a master timeline of the movie, an interval list for a node ~~defining~~including, for each ~~possible time in the range~~interval in the node's timeline, a single editing timestamp ~~corresponding to the possible time~~;

update the interval list for a node when the node is edited ~~by adding an interval corresponding to the period of time affected by the edit and having as its timestamp the timestamp of the edit~~; and

use the interval list for a first node to evaluate the validity of a cached frame for a particular ~~frame time period~~interval of the master timeline, the cached frame having been produced by compositing ~~a~~the first node in the compositing tree, the evaluation being performed by (a) comparing (i) an editing timestamp associated with the cached frame with (ii) the editing timestamps of ~~all~~ intervals in the interval list that ~~overlap the frame time period~~map to at least a portion of the interval of the master timeline, and (b) treating the cached frame as invalid if any of the intervals' editing timestamps is later than the editing timestamp ~~of~~associated with the cached frame.

2. (Original) The product of claim 1, ~~wherein the product~~ further comprising instructions to:

use the interval lists of all nodes below the first node in the tree to evaluate the validity of the cached frame.

3. (Currently amended) The product of claim 1, wherein:

~~update~~updating the global timestamp comprises incrementing the global timestamp; and

the interval list is stored as a series of pairs (~~time~~interval, timestamp), the series being sorted by ~~time~~the intervals.

A1

4. (Currently amended) The product of claim 3, wherein:

the series of (~~time~~interval, timestamp) pairs ~~is~~ are sorted in ~~time~~ ascending order of intervals.

---

5. (New) A computer program product, tangibly stored on a machine-readable medium, for displaying a frame of a movie composition, the product comprising instructions operable to cause a programmable processor to:

A2

associate edit sequence information with an element of the movie composition, the edit sequence information specifying, for an interval of the element's timeline, an edit sequence position representing the position in a sequence of edits made to the movie composition of a most recent edit made that affects the element during the interval, the interval being a portion of the timeline;

when caching a frame, associate with the cached frame an edit sequence position that represents a state of editing of the movie composition; and

when displaying the frame, compare the edit sequence position associated with the cached frame with edit sequence information associated with the element.

6. (New) The product of claim 5, wherein:

the edit sequence position that represents a state of editing of the movie composition includes the edit sequence position of a most recent edit made to the movie composition.

7. (New) The product of claim 5, further comprising instructions to:

in response to an edit made to the movie composition, update the edit sequence information.

8. (New) The product of claim 7, wherein:

the instructions to compare include instructions to compare the sequence position associated with the cached frame with the updated sequence information.

9. (New) The product of claim 5, further comprising instructions to:

for an edit made to the movie composition, identify an interval of the element's corresponding timeline that may be affected by the edit.

A2

10. (New) The product of claim 9, wherein:

the identified interval is the maximum range during which the edit may affect the element.

11. (New) The product of claim 5, further comprising instructions to:

for an edit made, identify an interval of the element's corresponding timeline that is affected by the edit.

12. (New) The product of claim 5, wherein instructions to compare include instructions to:

identify the edit sequence position of the most recent edit from the sequence information associated with the element; and

compare the edit sequence position associated with the cached frame with the identified edit sequence position.

13. (New) The product of claim 5, wherein the sequence information associated with the element is placed into groups, the product further comprising instructions to:

identify the most recent edit sequence information for each group.

14. (New) The product of claim 5, wherein:

the edit sequence information includes an interval list, the interval list specifying, for

each interval of the element's timeline, the edit sequence position representing a position in a sequence of edits made to the composition of a most recent edit made that affects the element during the interval.

15. (New) The product of claim 14, wherein:

each interval of the interval list includes a start time; and

except for the last listed interval of the interval list, each interval of the interval list is delimited by its start time and the start time of the subsequent interval.

16 (New) The product of claim 15, wherein:

the movie composition has a master timeline to which the element's timeline maps, the master time line including a start time and a stop time;

the first interval listed in the element's interval list has a start time that precedes the start time of the composition's master timeline; and

the last interval listed in the element's interval list extends beyond the stop time of the composition's master timeline.

17. (New) The product of claim 15, further comprising instructions to:

for an edit, determine the start time and duration of an interval when the edit may affect the element, and define new intervals in the interval list if the interval list does not include an interval having the start time and duration of the determined interval, the new intervals being defined based on the start time and the duration of the determined interval.

18. (New) The product of claim 17, wherein:

the instructions to define new intervals include instructions to define new intervals such that the intervals in the interval list do not overlap.

19. (New) The product of claim 17, further comprising:

associate the edit sequence position of the edit with the determined interval.

20. (New) The product of claim 15, wherein:

the edit sequence position is represented by an integer.

21. (New) The product of claim 20, wherein:

the interval list of an element includes a first array and a second array that is parallel to the first array, the first array including start times and the second array including integers representing edit sequence positions.

A2

22. (New) The product of claim 14, further comprising instructions to:

when displaying the frame, identify the interval of the interval list that affects the frame and compare the edit sequence position associated with the cached frame with the edit sequence position listed in the interval list for the identified interval.

23. (New) The product of claim 14, further comprising instructions to:

maintain a first interval list for a first type of type of edits and a second interval list for a second type of edits.

24. (New) The product of claim 23, further comprising instructions to:

in response to an edit to the composition, identify one or more interval lists to update; and

update the identified interval lists.

In the abstract:

Please amend the abstract as follows:

Methods and apparatus, including computer program products, implementing and using techniques for determining validity of cached frames of a compositing hierarchy as a composition tree of a digital video composition is edited. The techniques use interval lists associated with nodes in a compositing tree to determine which cached frames of a composition hierarchy are guaranteed to be valid in the face of arbitrary edits to the hierarchy.

~~50025547.doc~~

In the specification:

Please replace the paragraph beginning at page 2, line 5 with the following amended paragraph:

A node is one element in a compositing tree and can be a comp~~. It can be~~, a layer, or an input source item, such as footage.

Please replace the paragraph beginning at page 5, line 14 with the following amended paragraph:

As shown in FIGS. 1a-c, a ~~methods~~ method for tracking cached frames employs a data structure that will be called an interval list, instances of which are attached to compositing tree nodes, and a series of algorithms for updating interval lists on the tree as edits are made. In this implementation, reception of an editing command causes an edit engine to update the affected interval lists. The interval lists are queried when the system needs to determine whether a cached frame is valid or needs to be re-rendered.

Please replace the paragraph beginning at page 8, line 3 with the following amended paragraph:

As shown in FIG. 1c, when a frame is later requested from the edited layer or comp, the system queries the interval lists for the range of time covered by that frame and ~~compare~~ compares it with the timestamp of a cached frame to determine whether the cached frame is valid or must be re-rendered. For example, if the system needs to obtain a particular frame of a compositing tree for a particular time in the animation timeline, the system determines if the frame is cached (decision step 122). If not, then the system renders the frame (step 123). If there is a cached version of the desired frame, then the system recursively searches the appropriate interval lists on the compositing tree to validate the cached version of the desired frame (step 124). A cached frame is considered valid unless its timestamp (a cached-frame timestamp or CFTS) is earlier than the timestamp of some interval in interval lists local to the node of the desired frame. If the search yields no interval timestamp later than the CFTS, then the cached frame is used (step 128).

Please replace the paragraph beginning at page 8, line 15 with the following amended paragraph:

In this implementation, the back end database of the application caches frames from two types of location in a composition tree. Multiple frames of the output of a comp are cached and each layer is cached. Layers are cached in a state representing the source pixels after the layers have been masked and effects have been applied, but before geometric transforms, lighting, and shading have been applied. This caching is referred to as the Post-Effect Cache or PFC. This feature allows the user ~~modifying~~to modify the geometric transform of a layer without incurring the cost of unnecessarily re-rendering the effects.

Please replace the paragraph beginning at page 8, line 28 with the following amended paragraph:

Comp and layer nodes have multiple interval lists, each representing time intervals and timestamps corresponding to editing operations that affect a particular subset of caches in a compositing tree. For example, operations such as translating and rotating a layer are grouped into an interval list for the layer because each operation causes corresponding cached frames of the parent comp to re-render, but not the PFC of the layer (unless the layer has a collateral dependency on the transformation). Having interval lists for different editing operations permits the compositing system to track individual frames of a node. Consequently, the compositing system advantageously has to re-render only some rather than all of the cached frames that depend on parameters of a node. For example, if a layer with effects described in the preceding paragraph is rotated, the PFC does not to be re-rendered. Only the comp needs to be recomposited using the pre-existing PFCs of each layer. The layer effects, assuming there is a corresponding PFC, ~~does not require re-rendering~~do not need to be rendered. Furthermore, grouping the hundreds of types of edit into a small number of interval lists saves storage space while advantageously permitting the tracking of individual frames of a node as discussed above. For example, if a layer is rotated and translated, the system needs to check only a single interval list to determine whether the cached frame of the layer is valid. However, if the translate and rotate operations each had an interval list, then the system would have to check them both to

validate a cached comp frame. Furthermore, there may occasionally be a cached frame that depends only on the translation and not the rotation because of collateral expression dependencies. This implementation groups these edit types into a single interval list.

Please replace the paragraph beginning at page 9, line 15 with the following amended paragraph:

The system has separate routines for the different levels at which frames are cached. Each routine specifies which interval lists to check. For example, if the system is determining the validity of a PFC, then only interval lists corresponding to editing operations that affect the masking and effects of a layer are considered. Interval ~~list~~ lists affecting the geometric transforms of the layer but not its effects are not checked unless there are collateral dependencies requiring their checking.

Please replace the paragraph beginning at page 10, line 7 with the following amended paragraph:

To see how interval list queries are used to decide the validity of a cached frame, consider, for example, that the system is called upon to render a frame at time T, and finds that a cached copy exists whose timestamp is M. If motion blur is not in use, the system will query the interval list with start time = T and duration equal to the comp's frame duration = D. (If motion blur is in use, that range will be expanded to included the full range of time at which the shutter is open, which may include time before or after the actual frame). FIG. 2 is a flow diagram of a method 200 of retrieving a comp frame. First, the system checks whether a cached comp frame is available (step 210). Then, the appropriate local interval lists of the comp are queried by calling a routine (e.g., process 400 of FIG. 4) for determining the validity of a comp frame (step 220). If the query returns a timestamp greater than M, the system knows immediately that the frame must be re-rendered (decision 230, and steps 250, 260, and 270), and no further queries are done. To re-render, the system clears the comp buffer (step 250) and calls a routine that retrieves the post mask and effects pixels (e.g., process 300 of FIG. 3). The system then transforms and composites the retrieved pixels (step 260). Additionally, the system caches the results and marks

the result with the current timestamp (step 270). However, if the called routine validates the cached comp frame by returning a "YES," process 200 causes the system to return the cached frame (step 240).

Please replace the paragraph beginning at page 10, line 24 with the following amended paragraph:

As shown in FIG. 3, to retrieve the post mask and effects pixels, the system calls a routine for determining the validity of the layer such as process 500 of FIG. 5 (step 310). If the called routine determines that the layer is valid and returns a "YES" (decision 320), then process 300 returns the cached image (step 330). If the called routine determines that the layer is invalid and returns a "NO," then the system re-renders (steps 360-380). To ren-render, the system determines if the source of the layer is a comp (decision 340). If the source is not a comp, then it is footage and the system retrieves the footage. The system applies masks and effects to the retrieved footage (steps 370 and 380). If the source is a comp, then the system calls a routine to retrieve the comp frame such as process 200 of FIG. 2 (step 360). The system applies masks and effects to the retrieved comp frame.

Please replace the paragraph beginning at page 11, line 3 with the following amended paragraph:

As shown in FIG. 4, in determining the validity of a comp frame, the system first queries the interval lists that affect comp output, i.e., those that directly affect comp pixels (step 410 and decision 420). If the cached frame timestamp is earlier than the corresponding timestamps in the interval lists queried, then the cached frame is invalid and the process returns a "NO" (step 460), thereby causing the system to re-render the frame in question. However, if the cached frame timestamp is not earlier than the corresponding timestamps, then the process checks each of the layers that are active at time T (step 430 and decision 440). This check is done for each layer by calling a routine such as process 500 of FIG. 5 (step 430). However, comp time T and duration D may not correspond to the same times in the layer, because the layer may be shifted and scaled in time. In this implementation, the interval (T, D) is mapped by the layer's time parameters. If

the called routine returns a "YES," thus validating the comp frame, process 400 returns a "YES" (step 450), thus in turn causing the system to use the valid cached comp frame. If the called routine returns a "NO," thus invalidating the cached comp frame, process 400 returns a "NO" (step 460), causing the system to re-render.

Please replace the paragraph beginning at page 11, line 27 with the following amended paragraph:

As shown in FIG. 6, the system queries the local interval lists that affect the PFC (step 605 and decision 610). If the system finds a timestamp later than the cached frame timestamp, the layer is invalid and process 600 returns a "NO" (step 650). Otherwise, the system checks collateral dependencies of the layer (step 615 and decision 620). If a timestamp later than the cached frame timestamp is found here, the layer is invalid and the process returns a 'NO" (step 650). However, if no later timestamps are found, the system checks whether the layer source is a comp or a footage (decision 625). If the source is footage, which has a single timestamp instead of an interval list, then the system simply compares the cached frame timestamp to the footage timestamp (step 640 and decision 645). If the footage timestamp is later than the cached frame timestamp, then the system returns a "NO" (step 650). Otherwise, the system ~~return~~ returns a "YES" (step 660). If the layer source is a comp, then the system calls a routine such as process 400 of FIG. 4 to determine the validity of the comp (step 630). If the comp is valid (decision 635), then the system returns a "YES" (step 660). Additionally, the system searches interval lists of collateral dependencies because changes to a collateral dependency affect the dependent layer. Each active layer of the original comp is checked in this way. If at the end of the recursion no later timestamps are found, the system may safely assume that the cached comp frame is up to date and may be used. Note that although the comp the system originally started with may be out of date and need re-rendering, one or more of its input nodes may be up to date, and their cached frames can be used in computing the final comp, thereby saving time.